

(11)Publication number : 2003-044093  
(43)Date of publication of application : 14.02.2003

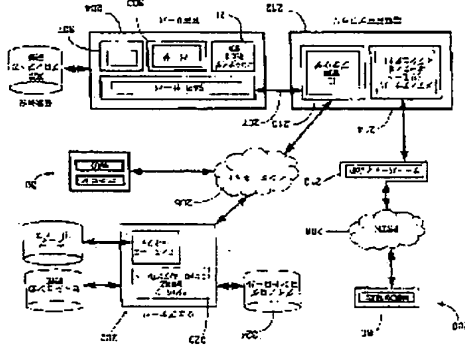
(11)特許出願公開番号  
特開2003-44093  
(P2003-44093A)

(21)Application number :	2002-132052	(71)Applicant :	MICROSOFT CORP
(22)Date of filing :	07.05.2002	(72)Inventor :	WANG KUANSAN HON HSIAO-WUEN

Priority number : 2001 289041	Priority date : 04.05.2001	Priority country : US
2001 960229	20.09.2001	US
2002 117141	05.04.2002	US

**PROBLEM TO BE SOLVED:** To enable a server for web enabled speech recognition, which is used for providing speech recognition in the server/client architecture of the Internet or the like, to have a unified architecture.

**SOLUTION:** A markup language for execution on a client device in a client/ server system contains an instruction for unifying at least one of events related to recognition, a GUI event and a telephony event on a non-display, voice input based client device 80 and a multimodal based client 30 for a web server 202 to interact with each of client devices. A recognition server 204 is provided as well for receiving data showing input data provided to the client device and an instruction of a grammar to be used for recognition.



## [Date of final disposal for application]

$$\frac{1}{1}$$

インクデバイスを日常活動で使用する頻度は増す一方である。現在では、こうしたデバイスを動作させるのに使用されるマイクプロセッサに利用できる処理能力が増大したことにより、これらデバイスの機能性が高まっており、場合によっては機能を一体化している。例えば現在、携帯電話の多くは、アドレス、電話番号などの個人情報

情報の記憶に使用できるだけでなく、インターネットのアクセスおよびブラウザにも使用することができる。

【0003】こうしたコンピュータデバイスをインターネットブラウザに使用し、あるいは他のサーバ/クライアントアーキテクチャで使用するところから、情報をコンピュータデバイスに入力することが必要となる。不都合なのは、移行を容易にするためにこうしたデバイスを可能な限り小さくしたいという要求があり、利用可能なコンピュータデバイス自体の表面積が限られているために、アルファベットの全文字を個別のボタンとして備える従来型のキーボードは適用は不可能であることである。

【0004】最近、VoiceXML (音声拡張可能マークアップ言語) の使用によるなどの音声ポータルが進歩し、電話だけを使用してインターネットコンテンツにアクセスすることが可能になっている。このアーキテクチャでは、ドキュメントサーバ (例えばウェブサーバ) が、VoiceXMLインタープリタを通じてクライアントからの要求を処理する。ウェブサーバはそれに応じてVoiceXMLドキュメントを生成することができ、このドキュメントはVoiceXMLインタープリタによって処理し、ユーザに対して音声としてレンダリングされる。ユーザは、音声認識を通じて音声コマンドを使用することにより、ウェブをナビゲートすることができ

る。

【0005】VoiceXMLは、フロー制御タグを用いるマークアップ言語であるが、フロー制御は、イベントینگ (eventing) および個別のスク립トを含むHTML (ハイパーテキストマークアップ言語) のフロー制御モデルには従わない。VoiceXMLは一般に、電話ベースの音声のみの対話に特に適したフォーム解釈アルゴリズムを含むが、このアルゴリズムでは通例、ユーザから得られる情報をシステムまたはアプリケーションによって制御する。グラフィカルユーザインタフェースも提供し、クライアントサーバ間で利用することのできるアプリケーションにVoiceXMLを直接組み込むには、開発者は、2つの形態のウェブページを習得する必要がある。すなわち、VoiceXMLのオーサリングと、HTML (など) を使用したオーサリングであるが、これらはそれぞれ異なるフロー制御モデルに従っている。

【0006】

【発明が解決しようとする課題】したがって、インターネットなどのサーバ/クライアントアーキテクチャで音声認識を提供するのに使用されるアーキテクチャ、またはその部分、および方法に改良を加えることが現在必要とされている。音声認識用のオーサリングツールは、PIM、電話などの小型のコンピュータデバイスに容易に適合できなければならない。前述の不利益の1つ、いくつか、またはすべてに対処するウェブオーサリングのアーキテクチャまたは方法が特に必要とされる。

【0007】本発明は、このような課題に鑑みとされ、たもので、その目的とするところは、インターネットなどのサーバ/クライアントアーキテクチャで音声認識を提供するのに使用される、統一したアーキテクチャを備えたウェブ対応音声認識用サーバの方法および記録媒体を提供することにある。

【0008】

【課題を解決するための手段】データ処理用のサーバ/クライアントシステムは、リモートにアクセスできる情報を含んだウェブサーバを有するネットワークを含む。クライアントデバイスは、マイクプロセッサなどの入力装置と、スピーカまたはディスプレイなどのレンダリング構成要素を含む。クライアントデバイスは、ウェブサーバから情報を入手して、その情報に含まれるフィールドと関連付けられた入力データを記録するように構成する。クライアントデバイスは、認識に使用する文法の指示とともに入力データを遠隔位置に送信するように適合する。

【0009】本発明の一態様として、認識サーバは入力データおよび文法の指示を受け取る。認識サーバは、何が入力されたかを示すデータをクライアントおよびウェブサーバの少なくとも1つに送す。

【0010】本発明の第2の態様として、クライアントサーバシステム中のクライアントデバイスで実行するマークアップ言語は、各クライアントデバイスと対応するウェブサーバのために、非表示式の音声入力ベースのクライアントデバイスとマルチモダルベースのクライアントにおける、認識に関連するイベント、GUIイベント、および電話イベントのうち少なくとも1つを統一する命令を含む。

【0011】

【発明の実施の形態】ウェブベース認識のアーキテクチャおよびその実施方法を説明する前に、このアーキテクチャで機能することが可能なコンピュータデバイスについて全般的に説明しておくことが有用である。本明細書で図1を参照すると、データ管理デバイス (PIM、PDAなど) の例示的形態が30に表されている。ただし、本発明は、下記で論じるこの他のコンピュータデバイス、特に入力ポタンなどを装備するには限

面が限られたコンピュータデバイスを使用して実施することも企図している。例えば、電話および/またはデータ管理デバイスも、本発明から利益を受けることができる。このようなデバイスは、既述の携帯個人

情報提供装置を含む。ローカルおよびリモートどちらのコンピュータ記憶媒体に置いてもよい。以下で、図面の助けを借りて、プログラムおよびモジュールによって実行するタスクを説明する。当業者は、この説明および図面をプロセッサ実行可能命令として実施することができ、この命令はどの形態のコンピュータ読み取り可能な記憶媒体にも書き込むことができる。

【0022】図4を参照すると、コンピュータ120の構成要素には、プロセッサ140、システムメモリ150

とができる。このようなデバイスは、既述の携帯個人

情報提供装置を含む。ローカルおよびリモートどちらのコンピュータ記憶媒体に置いてもよい。以下で、図面の助けを借りて、プログラムおよびモジュールによって実行するタスクを説明する。当業者は、この説明および図面をプロセッサ実行可能命令として実施することができ、この命令はどの形態のコンピュータ読み取り可能な記憶媒体にも書き込むことができる。

【0023】通信媒体は、通例、搬送波または他の搬送波などの変調データ信号中のコンピュータ可読命令、データ構造、プログラムモジュール、または他のデータを実施し、また任意の情報伝達媒体を含む。用語「変調データ信号」とは、情報を信号中に符号化するような方式で、その特性の1つまたは複数を変化または変化したワークまたは直接配線線路などの配線媒体、および音響、F R、赤外線、および他の無線媒体などの無線媒体が含まれるが、これらに限定しない。上記の媒体の任意の組合せも、コンピュータ読み取り可能な記録媒体の範囲に含まれるものとする。

【0024】最近、VoiceXML (音声拡張可能マークアップ言語) の使用によるなどの音声ポータルが進歩し、電話だけを使用してインターネットコンテンツにアクセスすることが可能になっている。このアーキテクチャでは、ドキュメントサーバ (例えばウェブサーバ) が、VoiceXMLインタープリタを通じてクライアントからの要求を処理する。ウェブサーバはそれに応じてVoiceXMLドキュメントを生成することができ、このドキュメントはVoiceXMLインタープリタによって処理し、ユーザに対して音声としてレンダリングされる。ユーザは、音声認識を通じて音声コマンドを使用することにより、ウェブをナビゲートすることができ

る。

【0005】VoiceXMLは、フロー制御タグを用いるマークアップ言語であるが、フロー制御は、イベントینگ (eventing) および個別のスク립トを含むHTML (ハイパーテキストマークアップ言語) のフロー制御モデルには従わない。VoiceXMLは一般に、電話ベースの音声のみの対話に特に適したフォーム解釈アルゴリズムを含むが、このアルゴリズムでは通例、ユーザから得られる情報をシステムまたはアプリケーションによって制御する。グラフィカルユーザインタフェースも提供し、クライアントサーバ間で利用することのできるアプリケーションにVoiceXMLを直接組み込むには、開発者は、2つの形態のウェブページを習得する必要がある。すなわち、VoiceXMLのオーサリングと、HTML (など) を使用したオーサリングであるが、これらはそれぞれ異なるフロー制御モデルに従っている。

【0006】

【発明が解決しようとする課題】したがって、インターネットなどのサーバ/クライアントアーキテクチャで音

【0018】上記の図式またはモジュール型のコンピュータデバイス以外に、本発明は、一般的なデスクトップコンピュータなど数多くの他のコンピュータデバイスにも使用できることが理解されよう。例えば、身体能力が限られたユーザにとって完全な英数字キーボードなどの従来の入力装置の操作が困難である場合には、本発明は、そのようなユーザがコンピュータまたは他のコンピュータデバイスにテキストを入力することを可能にする。

【0019】本発明は、数多くの他の利用または特殊目的のコンピュータシステム、環境、または構成での動作が可能である。本発明とともに使用するのに適した周知のコンピュータシステム、環境、および/または構成の例には、従来型の (regular) パーバコンピュータ、携帯用デバイスまたはラップトップデバイス、マルチプロセッサシステム、マイクロプロセッサベースのシステム、セットトップボックス、プログラマブル家庭用電化製品、ネットワークPC、ミニコンピュータ、メインフレームコンピュータ、上記のシステムまたはデバイスなどの任意のものを含む分散型コンピュータネットワーク環境が含まれるがこれらに限定するものではない。

【0020】以下は、図4に示す汎用コンピュータ120の簡単な説明である。ただし、この場合もコンピュータ120は、適切なコンピュータ環境の一例に過ぎず、本発明の使用または機能性の範囲に関して何らの制限を示唆するものではない。また、コンピュータ120は、この図に示す構成要素のいずれか、またはその組合せに関する依存性や要件を有するものとも解釈すべきではない。

【0021】本発明は、プログラムモジュールなどコンピュータで実行するコンピュータ実行可能命令の一般的な状況で説明することができる。一般に、プログラムモジュールには、特定タスクを実行する、または特定の抽象データタイプを定義する、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造などが含まれる。本発明はまた、通信ネットワークを通じてリンクした遠隔処理装置によってタスクを実行する分散型コンピュータデバイス環境でも実施することができる。分散型コンピュータデバイス環境では、プログラムモジュールは、メモリ記憶装置を含む、ローカルおよびリモートどちらのコンピュータ記憶媒体に置いてもよい。以下で、図面の助けを借りて、プログラムおよびモジュールによって実行するタスクを説明する。当業者は、この説明および図面をプロセッサ実行可能命令として実施することができ、この命令はどの形態のコンピュータ読み取り可能な記憶媒体にも書き込むことができる。

【0022】図4を参照すると、コンピュータ120の構成要素には、プロセッサ140、システムメモリ150

とができる。このようなデバイスは、既述の携帯個人

情報提供装置を含む。ローカルおよびリモートどちらのコンピュータ記憶媒体に置いてもよい。以下で、図面の助けを借りて、プログラムおよびモジュールによって実行するタスクを説明する。当業者は、この説明および図面をプロセッサ実行可能命令として実施することができ、この命令はどの形態のコンピュータ読み取り可能な記憶媒体にも書き込むことができる。

【0023】通信媒体は、通例、搬送波または他の搬送波などの変調データ信号中のコンピュータ可読命令、データ構造、プログラムモジュール、または他のデータを実施し、また任意の情報伝達媒体を含む。用語「変調データ信号」とは、情報を信号中に符号化するような方式で、その特性の1つまたは複数を変化または変化したワークまたは直接配線線路などの配線媒体、および音響、F R、赤外線、および他の無線媒体などの無線媒体が含まれるが、これらに限定しない。上記の媒体の任意の組合せも、コンピュータ読み取り可能な記録媒体の範囲に含まれるものとする。

【0024】最近、VoiceXML (音声拡張可能マークアップ言語) の使用によるなどの音声ポータルが進歩し、電話だけを使用してインターネットコンテンツにアクセスすることが可能になっている。このアーキテクチャでは、ドキュメントサーバ (例えばウェブサーバ) が、VoiceXMLインタープリタを通じてクライアントからの要求を処理する。ウェブサーバはそれに応じてVoiceXMLドキュメントを生成することができ、このドキュメントはVoiceXMLインタープリタによって処理し、ユーザに対して音声としてレンダリングされる。ユーザは、音声認識を通じて音声コマンドを使用することにより、ウェブをナビゲートすることができ

る。

【0005】VoiceXMLは、フロー制御タグを用いるマークアップ言語であるが、フロー制御は、イベントینگ (eventing) および個別のスク립トを含むHTML (ハイパーテキストマークアップ言語) のフロー制御モデルには従わない。VoiceXMLは一般に、電話ベースの音声のみの対話に特に適したフォーム解釈アルゴリズムを含むが、このアルゴリズムでは通例、ユーザから得られる情報をシステムまたはアプリケーションによって制御する。グラフィカルユーザインタフェースも提供し、クライアントサーバ間で利用することのできるアプリケーションにVoiceXMLを直接組み込むには、開発者は、2つの形態のウェブページを習得する必要がある。すなわち、VoiceXMLのオーサリングと、HTML (など) を使用したオーサリングであるが、これらはそれぞれ異なるフロー制御モデルに従っている。

【0006】

【発明が解決しようとする課題】したがって、インターネットなどのサーバ/クライアントアーキテクチャで音

声認識を提供するのに使用されるアーキテクチャ、またはその部分、および方法に改良を加えることが現在必要とされている。音声認識用のオーサリングツールは、PIM、電話などの小型のコンピュータデバイスに容易に適合できなければならない。前述の不利益の1つ、いくつか、またはすべてに対処するウェブオーサリングのアーキテクチャまたは方法が特に必要とされる。

【0007】本発明は、このような課題に鑑みとされ、たもので、その目的とするところは、インターネットなどのサーバ/クライアントアーキテクチャで音声認識を提供するのに使用される、統一したアーキテクチャを備えたウェブ対応音声認識用サーバの方法および記録媒体を提供することにある。

【0008】

【課題を解決するための手段】データ処理用のサーバ/クライアントシステムは、リモートにアクセスできる情報を含んだウェブサーバを有するネットワークを含む。クライアントデバイスは、マイクプロセッサなどの入力装置と、スピーカまたはディスプレイなどのレンダリング構成要素を含む。クライアントデバイスは、ウェブサーバから情報を入手して、その情報に含まれるフィールドと関連付けられた入力データを記録するように構成する。クライアントデバイスは、認識に使用する文法の指示とともに入力データを遠隔位置に送信するように適合する。

【0009】本発明の一態様として、認識サーバは入力データおよび文法の指示を受け取る。認識サーバは、何が入力されたかを示すデータをクライアントおよびウェブサーバの少なくとも1つに送す。

【0010】本発明の第2の態様として、クライアントサーバシステム中のクライアントデバイスで実行するマークアップ言語は、各クライアントデバイスと対応するウェブサーバのために、非表示式の音声入力ベースのクライアントデバイスとマルチモダルベースのクライアントにおける、認識に関連するイベント、GUIイベント、および電話イベントのうち少なくとも1つを統一する命令を含む。

【0011】

【発明の実施の形態】ウェブベース認識のアーキテクチャおよびその実施方法を説明する前に、このアーキテクチャで機能することが可能なコンピュータデバイスについて全般的に説明しておくことが有用である。本明細書で図1を参照すると、データ管理デバイス (PIM、PDAなど) の例示的形態が30に表されている。ただし、本発明は、下記で論じるこの他のコンピュータデバイス、特に入力ポタンなどを装備するには限

面が限られたコンピュータデバイスを使用して実施することも企図している。例えば、電話および/またはデータ管理デバイスも、本発明から利益を受けることができる。このようなデバイスは、既述の携帯個人

情報提供装置を含む。ローカルおよびリモートどちらのコンピュータ記憶媒体に置いてもよい。以下で、図面の助けを借りて、プログラムおよびモジュールによって実行するタスクを説明する。当業者は、この説明および図面をプロセッサ実行可能命令として実施することができ、この命令はどの形態のコンピュータ読み取り可能な記憶媒体にも書き込むことができる。

【0022】図4を参照すると、コンピュータ120の構成要素には、プロセッサ140、システムメモリ150

とができる。このようなデバイスは、既述の携帯個人

情報提供装置を含む。ローカルおよびリモートどちらのコンピュータ記憶媒体に置いてもよい。以下で、図面の助けを借りて、プログラムおよびモジュールによって実行するタスクを説明する。当業者は、この説明および図面をプロセッサ実行可能命令として実施することができ、この命令はどの形態のコンピュータ読み取り可能な記憶媒体にも書き込むことができる。

【0023】通信媒体は、通例、搬送波または他の搬送波などの変調データ信号中のコンピュータ可読命令、データ構造、プログラムモジュール、または他のデータを実施し、また任意の情報伝達媒体を含む。用語「変調データ信号」とは、情報を信号中に符号化するような方式で、その特性の1つまたは複数を変化または変化したワークまたは直接配線線路などの配線媒体、および音響、F R、赤外線、および他の無線媒体などの無線媒体が含まれるが、これらに限定しない。上記の媒体の任意の組合せも、コンピュータ読み取り可能な記録媒体の範囲に含まれるものとする。

【0024】最近、VoiceXML (音声拡張可能マークアップ言語) の使用によるなどの音声ポータルが進歩し、電話だけを使用してインターネットコンテンツにアクセスすることが可能になっている。このアーキテクチャでは、ドキュメントサーバ (例えばウェブサーバ) が、VoiceXMLインタープリタを通じてクライアントからの要求を処理する。ウェブサーバはそれに応じてVoiceXMLドキュメントを生成することができ、このドキュメントはVoiceXMLインタープリタによって処理し、ユーザに対して音声としてレンダリングされる。ユーザは、音声認識を通じて音声コマンドを使用することにより、ウェブをナビゲートすることができ

る。

【0005】VoiceXMLは、フロー制御タグを用いるマークアップ言語であるが、フロー制御は、イベントینگ (eventing) および個別のスク립トを含むHTML (ハイパーテキストマークアップ言語) のフロー制御モデルには従わない。VoiceXMLは一般に、電話ベースの音声のみの対話に特に適したフォーム解釈アルゴリズムを含むが、このアルゴリズムでは通例、ユーザから得られる情報をシステムまたはアプリケーションによって制御する。グラフィカルユーザインタフェースも提供し、クライアントサーバ間で利用することのできるアプリケーションにVoiceXMLを直接組み込むには、開発者は、2つの形態のウェブページを習得する必要がある。すなわち、VoiceXMLのオーサリングと、HTML (など) を使用したオーサリングであるが、これらはそれぞれ異なるフロー制御モデルに従っている。

【0006】

【発明が解決しようとする課題】したがって、インターネットなどのサーバ/クライアントアーキテクチャで音

声認識を提供するのに使用されるアーキテクチャ、またはその部分、および方法に改良を加えることが現在必要とされている。音声認識用のオーサリングツールは、PIM、電話などの小型のコンピュータデバイスに容易に適合できなければならない。前述の不利益の1つ、いくつか、またはすべてに対処するウェブオーサリングのアーキテクチャまたは方法が特に必要とされる。



【0056】認識を中止またはキャンセルすることができ、他の属性には、「bubbleTimeout」属性295があるが、これは285の音声の検出後に認識サーバ204が結果を返さなければならない期間である。この期間を超えると、エラー発生の有無に応じて異なるイベントが発生される。例えば、例外的に発音が長引いている場合は、「onNoRecol」属性293を実行する。しかし他の何らかの理由で「bubbleTimeout」属性295を超えた場合は、認識エラーの可能性が高くなり、「ontimeout」イベント297が発生される。同様に「maxTimeou」属性299も提供することができる。これは、認識の開始283から結果をクライアント30に返すまでの期間である。この期間を超えると、「ontimeout」イベント297が発生される。

【0057】ただし、「endSilence」属性301以上の期間を超えた場合、これは認識が完了していることを示唆するが、この場合は認識サーバ204が自動的に認識を中止し、その結果を返す。認識サーバ204は、信頼度の測定を実施して、認識結果を返すべきかどうかを判定できることに留意されたい。信頼度の測定値が閾値を下回る場合は、「onNoRecol」属性293を発行し、一方信頼度の測定値が閾値を上回る場合は、「onNoRecol」属性303および認識結果を、「onNoRecol」属性303に提供することができる。したがって図14は、「自動モード」で、明示的なstop( )の呼び出しが行われていない状況を表している。

【0058】図15は、認識サーバ204の「シングルモード」の動作を図式的に表したものである。「自動モード」の関連で上記で説明した属性およびイベントを適用することができ、したがって同じ参照番号で示している。しかし、この動作モードでは、stop( )呼び出し305を、スケジュール281上に表示している。stop( )呼び出し305は、ユーザによる「ペンアップ」などのイベントに相当する。この動作モードでは、認識結果を返すことは、明示的なstop( )呼び出し305によって制御される。すべての動作モードの場合と同じく、「onSilence」イベント291は、「initialTimeout」期間289内で音声が検出されない場合に発生するが、この動作モードでは認識を中止しない。同様に、stop( )呼び出し305以前の認識不可能な発声によって生成される「onNoRecol」イベント293によっても認識は中止されない。ただし、「bubbleTimeout」属性295または「maxTimeou」属性299と関連付けられた期間を超えた場合は、認識を中止する。

【0059】図16は、認識サーバ204の「複数モード」の動作を図式的に表している。上記で指摘したよう

に、この動作モードは、「オープンマイクプロフォン」またはディクテーションのシナリオで使用される。一般に、この動作モードでは、明示的なstop( )呼び出し305が受け取られるか、または「bubbleTimeout」属性295または「maxTimeou」属性299に関連付けられた期間を超えたまで、間隔を置いて認識結果を返す。ただし、「onSilence」イベント291、「onRecol」イベント303、または「onNoRecol」イベント293のいずれかが発生すると、これらによって認識は中止されないが、「bubbleTimeout」期間および「maxTimeou」期間のタイマがリセットされることに留意されたい。

【0060】一般に、この動作モードでは、stop( )呼び出し305が受け取られるまで、認識されるフレーズごとに、「onRecol」イベント303を発行し、結果を返す。認識不可能な発声のために「onSilence」イベント291が発生された場合は、これらのイベントを報告するが、認識は継続する。【0061】上記で触れたように、フィールドに関連付けられた1つまたは複数のrecognitionオブジェクトを起動するが、これには、少なくとも1つの文法を使用するに付いての指示を認識サーバ204に提供することができる。この情報は、クライアント30で記録して認識サーバ204に送信した音声データを伴うことができる。上記で指摘したように、音声データは、ユーザが入力した音声に関連づけられたストリーミングデータを含むことができ、あるいは音声認識中に使用する音声の特徴を示す、前処理済みの音声データを含むことができる。別の実施形態では、クライアント側の処理に音声データの正規化も含むことができ、認識サーバ204が受け取る音声データが、クライアントごとに比較的時間になるようにする。これにより認識サーバ204の音声処理が簡略化され、認識サーバを、クライアントおよび通信経路のタイプにステータスにすることができるので、認識サーバ204のスケラビリティをより容易にすることができる。

【0062】認識サーバ204から認識結果を受け取ると、その認識結果を対応するフィールドと関連付け、必要な場合はクライアント側で確認またはチェックを行うことができる。現在クライアントがレンダリングしているコードと関連付けられたすべてのフィールドを完了すると、アプリケーション処理のためにその情報をウェブサーバ202に送信する。前述の内容から、ウェブサーバ202は、認識に適したコードまたはページ/スクリプトをクライアント30に提供しているが、認識サーバ204によって行われることは明白である。ただし、本発明は、認識サーバ204をウェブサーバ202とまとめて配置する、または認識サーバ204をクライ

(incur)ことなく、そのページのオブジェクトのメソッドを起動することができる。【0075】上記の実施形態では、「bind」要素は、認識結果をフォーム中またはウェブページ中のフィールドに割り当てるための属性「targetElement」および「targetAttribute」しがある。別の実施形態では、「bind」要素は、オブジェクトメソッドの起動のために加える「targetMethod」も含む。「targetMethod」の使用および機能は、スク립ティングの領域にとって非常に重要な技術である。例えば、次の構文を使用して、オブジェクト「OBJ1」の「X」メソッドを起動することができる。

```
<bind TargetElement = "OBJ1" TargetMethod = "X"
...>
```

ここに示す例はHTML/XHTMLのイベント構文に従っているが、当業者にとっては、<bind>の使用を一般化して、他のイベントリング機構を使用することを平易であることに留意されたい。他のイベントリング機構には、W3Cドキュメントオブジェクトモデルレベル2またはレベル3のイベントリング規格、ECMA共通言語仕様(CLI)イベントモデル、Java(登録商標)プログラミング言語イベントモデル、W3C同期マルチメディア統合言語(SMIL)、および近々登場するW3CのXMLイベント規格提案が含まれるが、これらに限定するものではない。

【0076】図17および18は、クライアント、特にダウンレブルブラウザで実行可能なマークアップ言語のページである。この例では、音声プロンプトを通じてユーザに希望する飲料を尋ねている。このシステムは次の飲料が注文されたかを確認する。認識結果に応じて、「bind」要素は、宣言した論理を使用して実行を導く。飲料を承認すると、そのフォームをウェブサーバ202に再度提出するが、これらにスク립ティングは一切用いない。

【0077】一般に、図17および18のマークアップ例は、データ部分350、音声部分352、およびユーザインタフェース部分354、356、および358を含む。部分354は、全般的な質問から、ユーザが希望する飲料についての認識結果を受け取り、初稿式論理を誘導して、クリームや砂糖が必要かどうかについて再度指示を促し、尋ねるか、または注文された飲料を承認する。詳細には、部分356は、クリームや砂糖が注文された場合にはその認識結果を受け取る。部分358は、飲料の確認についての認識結果を受け取る。部分360は、新しいメッセージングオブジェクト「SMEX」を用いる制御部分である。「SMEX」については下記でさらに説明する。

【0078】上記で指摘したように、本発明のこの態様の「bind」要素はオブジェクトメソッドの起動を含む

expiry\_date)と関連付けられた情報を含む。この例では、電話音声ブラウザ212は、「onRecol」として認識済みの音声を受け取り、電話機80から受け取った音声データと、「domain」属性文法の使用の指示を認識サーバ204に送信し、関数「handle」が呼び出され、または実行されるが、これは音声データから認識されたフィールドの一部またはすべての値を関連付けることが含まれる。すなわち、認識サーバ204から得る結果は、各フィールドについての指示も含み得る。この情報は構文解析し、405で指定されるバインド規則に従って対応するフィールドと関連付ける。図5に示すように、認識サーバ204はパラサ309を含むことができる。

【0071】図7、8、9、10、11、および12から、非常に類似したウェブ開発フレームワークを使用する。データの提示も、これらの場合で非常に類似している。さらに、データ提示とフロー制御を分離することにより、異なるアプリケーション(システム主導型と組合主導型)間、または異なるモダリティ間(GUIウェブベース、音声のみ、およびマルチモダリティ)での再使用性を最大限にすることができる。また、これにより、電話機がディスプレイおよびデバイス30と同様の機能を含む場合に、音声のみの動作から電話、そしてマルチモダリティ動作への自然な拡張が可能になる。付録Aで、以上で説明したコンローラおよびオブジェクトの詳細にさらに提供する。

【0072】上記で指摘したように、アップレブルブラウザは、上記の例で認識結果を割り当てるために関数「handle」を起動するなど、各種のニーズを実行するためにスク립ティングを使用することができる。上記で説明し、付録Aの2.1.2にさらに説明する実施形態では、「bind」要素は認識結果を構文解析し、値を割り当てることが、この「bind」要素は「reco」要素の下位要素または子要素である。

【0073】スク립ティングは有用でありうるが、多くの者は、例えばセキュリティ問題などから必ずしも最も良のブラウザが実装形態であるとは限らないと見ている。したがって、本発明のさらに別の実施形態または態様では、「bind」要素は「reco」(同様の)高レベル要素であり、他のより豊富なプロパティとともに提供され、実際、それ自体ではスク립ティングを用いずにスク립ティングを実際に構築することができる。

【0074】スク립ティングを用いない場合、あるいは下記で述べる本発明の態様を使用しない場合、高度なダイアログ効果など下記で述べる機能の一部は、ページを再度ウェブサーバ202に提出し、そこでアプリケーションロジックを実行して新しいページを生成し、そのページを再びクライアントデバイスに送信することによってのみ実現することができる。本発明のこの態様により、プログラマは、サーバへのラウンドトリップを招く

オブジェクト（ここでは「smex」）と接す。例えば、ライアントデバイスで実行するアプリケーションから、18の例では、クライアントデバイスで実行される電話アプリケーションが呼を出すと、このページが実行される。部分360で、「bind」要素は、メッセージ「GaiCal\_connected」を受け取る、「welcome」プロンプトを実行または再生し、「redo\_dring」オブジェクトを再行することにより認識を開始する。音声サーバ204から受け取る認識結果と同様に、受け取るメッセージも大きく異なる可能性がある。メッセージの一部は、希望のプログラムフローを開けるために明確に決定する。受け取って処理することのできるメッセージもある（例えば、認識サーバから受け取る認識結果と同様に構文解析を行う）。

例えば、これにより、キーボードから入力するテキストの自然言語処理のようにクエリを実行できるようなものになる。付録Aの要素は、この機能を実行するためのプロパティを含んでいる。同様に、プロンプト要素を使用し、付録Aでさらに説明するプロパティ「innerlex」を使用することにより、動的コンテキストまたはオーディオウェブファイル用のテキストメッセージを提供することができ、イベントは、認識結果のためのイベントと同様のものではない。

例えば、イベントリングは「onReceived」を含むことができるが、これは、メッセージソース（例えばクライアントデバイスで実行するアプリケーション）が、ブラウザで使用できるメッセージを有する際に送られる。

【0087】このように、「smex」すなわちメッセージオブジェクトにより、ここに述べたようなマークアップタグを、クライアントデバイスで実行される他のコンポーネントまたはアプリケーションに拡張することが可能になる。例の例として、このメッセージオブジェクトを使用して、クライアントデバイスで実行される閲覧器書き手のTYPコンポーネントと通信することができは、TYPコンポーネントは、音声認識を使用するのである。ユーザが入力した内容のメッセージを提供する。このメッセージはその、認識結果を認識サーバへから受け取った場合と同様に使用する。すなわち、メッセージを構文解析して、フォームのフィールドに割り当てらる、あるいは上記の「reco」、「grammar」または「bind」要素を使用してその処理を行うことができる。このメッセージまたは「smex」オブジェクトについては、付録Aさらに説明する。

【0088】「bind」要素は「for」属性も含むことができ、これにより、その動作をページ上の他のオブジェクトに付することができる。例えば次のマークアップ

<kind for = "prompt" event = "onComplete" targetF 50 以下のタグは、ドキュメントが音声を入力媒体または出

-15-

lenant = "prompt2" = targetMethod = "start")>

は、オブジェクト「prompt 1」がイベント「prompt 2」のstartメソッドを起動する。

[0089] 再び図5を参照すると、ウェブサーバ20は、サーバ側のプラグイン宣言オースラリングツールすなわちモジュール320を含むことができる（例えば、あるイマイトリアル社によるASPまたはASP+、あるいはJSPなど）。サーバ側のプラグインモジュール320は、クライアント側のマークアップと、さらにはウェブサーバ202にアクセスするクライアントのタイプについて固有形態のマークアップ自動的に生成することができ、クライアント情報は、クライアント/サーバ関係が最初に確立されたときにウェブサーバ202に提供することができる。ウェブサーバ202は、クライアントの情報を検出するモジュールまたはルーチンを含むことができる。この方式で、サーバ側のプラグインモジュール320は、それぞれの音声認識ナリオ、すなわち電話機330を通じて音声のみ、あるいはマルチメディア型のテキスト330に対する、クライアント側のマークアップを生成することができる。一意性のあるクライアント側データを使用することにより（各アプリケーションで使用するクライアントおよびプロトコルコントロール）、使用されるクライアントのアプリケーションオースラリングが大幅に容易になる。

【0090】 クライアントマークアップの動的な生成に加えて、図8、9および10のマークアップ例を用いた、図8に示すようなクレジットカード番号の入手などの高レベルのダイアログモジュールは、アプリケーションオンセーリングの開発者が使用するために、記憶装置322に記憶するサーバ側コントロールとして実施することとができる。一般に、高レベルダイアログモジュール324は、開発者が指定するパラメータに基づいて、音声のみおよびマルチモジュールの両シリオで、クライアント側のマークアップおよびスクリプトを動的に生成する。高レベルダイアログモジュールは、開発者のニーズに適合するクライアント側のマークアップを生成するために、高レベルダイアログモジュールは、クレジットのマークアップを含むことができる。例えば、クレジットのカード情報のモジュールは、クライアント側のマークアップスクリプトが許可すべきクレジットカードの挿入を指定するパラメータを含むことができる。サーバ側プログラムのモジュール320で使用するASP+ページの例を図13に示す。

【0091】本發明について好ましい実施形態を参照して説明したが、当業者は、本発明の趣旨および範囲から逸脱せずに、形態および詳細を変更することが可能であることを理解されよう。

100921 付録A

## 一、概要

50 以下のタグは、ドキュメントが音声を入力媒体または出

は、認識結果を指定するためのセマンティックマークアップ言語 (SML) を含むXMLドキュメントでよい。その内容は、意味値、話された単語の単語 および信頼度スコアを含む。SMLは、代替の認識選択版 (N番目により認識結果におけるものなど) も含むことができ (sml confidence="40")

```
<travel text="シアトルから"
  <origin_city confid="1"
    </origin_city>
  <dest_city confid="1"
    </dest_city>
  </travel>
</smi>
```

**[0105]** 文法中 (in-grammar) 製版は、セマンティックマークアップ言語すなわちSMLでXMLドキュメントを生成することになっているので、SMLドキュメントからバイナリする場合は、XPathエディラを使用して参照する。また、値をバイナリするページ中の要素（これにはフォームコメントロールである可能性が高い）は一意に識別すべきなので、これらのタグゲット要素は直接参照する。

**【10106】属性:**  
・targetElement: 必須。SMLからvalueの内容を割り当てる要素 (W3C SMIL2.0と同様)。  
**【10107】・targetAttribute:** 任意。選択。SMLからvalueの内容を割り当てるターゲット要素の属性。(SMIL2.0のattribute Name属性と同様)。指定しない場合は、value 1 になる。

【0108】・test：任意選択。認識結果を割り当てる際の条件を示すXML Pattern (W3C XML DOM仕様と同様) ストリング。デフォルト条件は真。

【0109】・value：必須。ターゲット要素に割り当てる認識結果ドキュメントの値を指定するXPAT H (W3C XML DOM仕様と同様) ストリング。

【0110】例：上記のSMLのリターンを与えられる以下、以下の要素はパイナドで使用して、origin\_u\_cityおよびbest\_u\_city中の値を、ターゲットページの要素txtBoxOriginおよびtxtBoxDestに転送する。

\* 著者「I'd like to travel from Seattle to Boston (シアトルからボストンまで行きたい)」に対するSMLドキュメントの例を下に示す。

```
*
らボストンまで行きたい">
ence="45"> シアトル

ence="35"> ボストン

20
<input name="x1BoxOrigin" type="text"/>
<input name="x1BoxDest" type="text"/>

<recd id="travel">
  <grammar src=".city.xml" />

  <bind targetElement="x1BoxOrigin">
    value="//origin_city" />
  <bind targetElement="x1BoxDest">
    value="//dest_city" />
</recd>
```

このバインドは、バインド操作の事前条件として `des_t_cility` 結果の信頼度属性にテストを行う以下の例のように条件付きの場合もある。

```

30      <bind targetElement="textBoxDest"
      value="//dest_city"

```

test="/sm/desl\_city@confidence 8g1 40j

40 【01111】 2. 2 属性およびプロパティ  
以下の属性はすべてのブラウザでサポートされ、プロパティはアププレブルブラウザによってサポートされる。

【01122】 2. 2. 1 属性  
以下のRecの属性は、ダイアログターンのために音声レコグナイザを構成するために使用する。

【0113】 initialTimeout: 任意選  
択。認識の開始から音声の検出までのミリ秒単位の時間。この値は認識プラットフォームに渡され、これを超過した場合、onSilenceイベントが認識プラットフォームから提供される。(2 4 2参照) 指定された場合、

50

\* [0131] 2. 3. 4 Activate

Activateメソッドは、文脈自由文法 (CFG) の最上位規則を起動する。起動は、「開始された」認識プロセス中には効果を持たないで、認識が開始する前に呼び出さなければならぬ。明示的に非活性化していない認識コンテキストについてのすべての文法の最上位規則は、すでにアクティブであると見なすことに留意されたい。

[0132] 構文: Object. Activate  
10 (strName)

パラメータ:

・ strName: 必須。起動する規則名。

戻り値: なし

例外: なし

[0133] 2. 3. 5 Deactivate

このメソッドは、文法中のトップレベル規則を非活性化し、その規則が存在しない場合、このメソッドは効果を持たない。

構文: Object. Deactivate (strName)  
20

パラメータ:

・ strName: 必須。非活性化する規則名。空ストリングはすべての規則を非活性化する。

戻り値: なし

例外: なし

[0134] 2. 4 Recoイベント  
Reco DOMオブジェクトは以下のイベントをサポートし、そのハンドラはreco要素の属性として指定することができる。

[0135] 2. 4. 1 onReco: このイベントは、レコグナイザが、そのブラウザで利用することのできる認識結果を得ると起動される。自動モードのrecoの場合、このイベントは認識プロセスを自動的に中止し、リソースをクリアする (2. 3. 2参照)。onRecoは通例、認識結果のプログラムの分析と、ページ中への結果の使用に使用される。

[0136] 構文:

[0137]

[表1]

\* 40

インラインHTML	<Reco onReco="handler">
イベントプロパティ	Object.onReco = handler;
	Object.onReco =
	GetRef("handler");

[表2]

[0138] イベントオブジェクト情報:

[0139]

はデータについてイベントオブジェクトに照会を行うことができる。

[0148] 2. 4. 3 onTimeout  
onTimeoutは、通例は音声プラットフォームからのエラーを反映する2タイプのイベントを扱う。

[0149] ・認識が完了する前にmaxTime属性で指定された期間を過ぎた (2. 2. 1参照) ことを通

知する、タグインタープリタが投入するイベントを扱

う。このイベントは通例、分岐型アーキテクチャで生じ

うる問題を反映する。

\* 10

インラインHTML	<Reco onTimeout="handler">
イベントプロパティ	Object.onTimeout = handler;
(ECMAScript)	Object.onTimeout =
	GetRef("handler");

[0154] イベントオブジェクト情報:

[0155]

\* 40

バブル	なし
転送するには	認識の中止前に、maxTime 属性で指定された期間が過ぎるとブラウザが投入する
デフォルトアクション	reco ステータスを-12にセットする

[0156] イベントプロパティ: イベントハンドラは ★ドラである。それが発生しうる異なるケースは、ステータスコードで区別する。このイベントは認識プロセスを自動的に中止する。

[0157] 2. 4. 4 onNoReco: onNoRecoは、有効な認識結果を戻すことができない際に音声認識プラットフォームが投入するイベント用のハンドラ★

[0158] 構文:

[0159]

[表7]

インラインHTML	<Reco onNoReco="handler">
イベントプロパティ	Object.onNoReco = handler;
	Object.onNoReco =
	GetRef("handler");

[0160] イベントオブジェクト情報:

[0161]

[表8]

・ **element** の必要な属性を指定することができる。これは、例えば、HTML フォームコントロール中の値をデリファレンスするのに有用である。下の例では、`textBoxOrigin` 要素および `textBoxDest` 要素の `value` 属性を、プロンプトの出力前にテキストに代入している。

```
(prompt id="confirm")
  あなたが行きたいのは
  (value targetBleat="textBoxOrigin"
  targetAttribute="value" /)
  から
  (value targetBleat="textBoxDest"
  targetAttribute="value" /)
  ですか?
(prompt)
```

【0172】3. 1. 3 オーディオファイル \*  
(prompt)  
ピーツという音がしたらメッセージを録音してください  
(value href="/wav/beep.wav" /)  
(prompt)

【0173】3. 1. 4 参照プロンプト  
インラインの内容を指定する代わりに、`src` 属性を空要素とともに使用し、URI を介して外部の内容を参照することができる。例えば、

```
(prompt id="welcome"
src="/ACWelcomePrompts/welcome" /)
```

`src` 属性の対象は、インラインプロンプトに指定する上記の内容の任意部分またはすべてを保持することができる。

【0174】3. 2 属性およびプロパティ  
このプロンプト要素は、以下の属性 (ダウンロード、アップロード) およびプロパティ (ダウンロードおよびアップロード) を保持する。

【0175】3. 2. 1 属性  
・ `atts` : 任意選択。テキストから音声への合成用のマークアップ言語タイプ。デフォルトは「SAPI 5」。

【0176】・ `src` : インラインプロンプトを指定する場合は任意選択。参照するプロンプトの URI (3. 1. 4 参照)。

【0177】・ `bargein` : 任意選択。整機。プロンプトの開始から、人間の発音が再生を中断できるようなまでのミリ秒単位の時間。デフォルトは無限。すなわちバーgeinを許可しない。 `bargein=0` にすると、即時のバーgeinが可能になる。これは、ブラウザフォアグラウンドするなどの種のバーgeinにも該当する。 `recognition` を開始しどちらを使用可能にするかに応じて、キーワードまたはエネルギーベースのバーgein時間とこの方式で構成することができる。

50

```
//
</script>
<script for="window" event="onload">
<!--
  news.Start 0; // keyword.Start 0;
  //
</script>
</head>
<body>
```

```
<prompt id="news" bargein="0">
```

水曜日の株式市場も、投資家が、米連邦準備理事会の会合に先立ち大きな動きにつながる材料を得られなかったことから底堅に活気がありませんでした。ハイテク銘柄中心のナスダック総合指数は42.51ポイント下落し、2156.26で取引を終えました。ダウジョーズ工業平均株価は、午後に入って反騰がなく17.05ポイント下落して10866.46で取引を終えました。

【0193】3. 4. 1 onBookmark  
合成ブックマークに遭遇すると発生する。このイベントは再生を一時停止しない。

【0194】構文:

【0195】

【表9】

20

```
(prompt)
(reco id="keyword")
reject="70"
onReco="checkWbBargein()"
(grammar
src=http://denail/news bargein grammar.xml /)
</reco>
</body>
</html>
```

【0192】3. 4 プロンプトイベント \*

インラインHTML	<code>&lt;prompt onBookmark="handler"&gt;</code>
イベントプロパティ	<code>Object onBookmark = handler</code>
	<code>Object onBookmark =</code>
	<code>GetRef(handler)"</code>

【0196】イベントオブジェクト情報: ※【表10】

【0197】

バブル	なし
起動するには	レンダリングしたストリング中のブックマークに遭遇する
デフォルトアクション	ブックマークストリングを戻す

【0198】イベントプロパティ: イベントハンドラは イベントを構成するかの決定はプラットフォームによるプロパティを直接受け取ることはないが、ハンドラはデータについてこのイベントオブジェクトに照会を行うことができる。

【0199】3. 4. 2 onBargein: ユーザ

のバーgeinイベントを抽出すると発生する。(例えば

【表11】

【表11】

```
51
]]</script>
<body>
  <input name="textBoxOrigin" value="Seattle"
    type="text"/>
  <input name="textBoxDest" type="text" />
  ...
  <prompt id="confirm" onBargein="interrupt0"
    bargein="0">
    <bookmark mark="mark_origin_city" />
    <value targetElement="origin"
      targetAttributes="value" />から
    <bookmark mark="mark_dest_city"
      />行きたい先地を言って下さい
    </prompt>
    <reco onReco="ProcessCityConfirm0">
      <grammar src="/gm/1033/cities.xml" />
    </reco>
    ...
  </body>
</reco>
</body>
```

【0213】4 DTMF  
DTMF認識オブジェクトを作成する。このオブジェクトは、インラインのマークアップ電機構文を使用して、あるいはスクリプト中にインスタンス化することができる。起動すると、DTMFにより、プロンプトオブジェクトがページインイベントを発生させることができる。下記でDTMFとの関連で説明するタグおよびイベントインデックス、および第5で説明する呼び出し。一般には、音声ブラウザが216とメディアサーバ214間の対話に関連するものであることに留意されたい。

【0214】4.1 内容  
・dtmfgrammar：インライン文法  
・bind：DTMFの変換結果を適切なフィールドに割り当てる

属性：  
・targetElement：必須。部分的な認識結果を割り当てて必要な要素（参照：W3C SMIL2.0に同じ）。  
・targetAttribute：認識結果を割り当てて必要な要素の属性（参照：SMIL2.0に同じ）。デフォルトはvalue。  
・test：割り当ての条件。デフォルトは真。

【0215】

```
<DTMF id="extension" numDigits="7">
  <bind targetElement="phone_number" />
</DTMF>
<DTMF id="area_code" numDigits="3">
  <bind targetElement="area_code" />
</DTMF>
```

この例は、いかにしてユーザが敬称フィールドに入力するのを可能にするかを示している。  
【0217】例3：音声入力およびDTMF入力をとる  
50 に許可し、ユーザがDTMFを開始した際に音声を使用

4.4.1 onkeypress  
DTMFキーを押すと発生する。これは、HTMLコントロールから継承したデフォルトイベントを上書きする。ユーザがエスケープキーを押すと、onKeyPress\*  
\*essではなくonRecイベントが発生する。  
【0231】構文：  
【0232】  
【表15】

インラインHTML	<DTMF onKeyPress="handler">
イベントプロパティ	Object.onKeyPress = handler
	Object.onKeyPress =
	GetRef("handler");

【0233】イベントオブジェクト情報：  
【0234】

※

バブル	なし
起動するには	タクトーン電線のキーパッドを押す
デフォルトアクション	押されているキーを戻す

【0235】イベントプロパティ：イベントハンドラは  
プロパティを直接受け取ることはないが、ハンドラはデータについてこのイベントオブジェクトに照会を行うことができる。  
【0236】4.4.2 onReco 20  
DTMFセッションを終了すると発生する。このイベント★

インラインHTML	<DTMF onReco="handler">
イベントプロパティ	Object.onReco = handler
	Object.onReco =
	GetRef("handler");

【0239】イベントオブジェクト情報：  
【0240】

☆

バブル	なし
起動するには	ユーザがエスケープキーを押す、またはキーストロークの回復が指定の直を渡す
デフォルトアクション	押されているキーを戻す

【0241】イベントプロパティ：イベントハンドラは  
プロパティを直接受け取ることはないが、ハンドラはデータについてこのイベントオブジェクトに照会を行うことができる。  
【0242】4.4.3 onTimeout  
タイムアウトまでに、句の終了イベントを受け取らない◆

インラインHTML	<DTMF onTimeout="handler">
イベントプロパティ	Object.onTimeout = handler
	Object.onTimeout =
	GetRef("handler");

【0245】イベントオブジェクト情報：  
【0246】

【表20】



<p>(31)</p> <p>59</p> <p>い場合もある)。</p> <p>【0250】5. 2. 6 Record ユーザオーディオをファイルに記録する。</p> <p>【0260】構文: telephone. Record (url, endSilence, [maxTimeOut], [initialTimeOut]);</p> <p>パラメータ:</p> <ul style="list-style-type: none"> <li>・url: 必須。記録された結果のURL。</li> <li>・endSilence: 必須。無音の検出後に記録を中止するミリ秒単位の時間。</li> <li>・maxTimeOut: 任意選択。記録を行う秒単位の最大時間。デフォルトはプラットフォーム固有になる。</li> <li>・initialTimeOut: 任意選択。記録の開始時に検出される無音の最大時間 (ミリ秒)。</li> </ul> <p>戻り値: なし</p> <p>例外: 記録をURLに書き込めない際に例外を投入する。</p>	<p>(33)</p> <p>60</p> <p>【0261】5. 3 イベントハンドラ</p> <p>電話音声ブラウザを使用するアプリケーション開発者は、以下のイベントハンドラを実装することができる。</p> <p>【0262】5. 3. 1 onIncoming ()</p> <p>音声ブラウザが着信電話呼を受信すると呼び出される。すべての開発者は、電話呼に応答する前にこのハンドラを使用して発呼者のアドレスを読み取り、カスタマイズした機能を起動することができる。</p> <p>【0263】5. 3. 2 onAnswer ()</p> <p>音声ブラウザが着信呼に応答すると呼び出される。</p> <p>【0264】5. 3. 3 onHangup ()</p> <p>ユーザが電話を切ると呼び出される。このイベントは、プログラムがHangupメソッドまたはTransferメソッドを呼び出して自動的には発生しない。</p> <p>【0265】5. 4 例</p> <p>この例は、電話セッションを操作するために制御イベントに結合 (wire) したスク립ティングを示す。</p>	<p>特開2003-44093</p> <p>60</p>
<p>(32)</p> <p>61</p> <p>【0266】5. 4. 1 例</p> <p>この例は、電話セッションを操作するために制御イベントに結合 (wire) したスク립ティングを示す。</p>	<p>(33)</p> <p>62</p> <p>【0266】5. 4. 1 例</p> <p>この例は、電話セッションを操作するために制御イベントに結合 (wire) したスク립ティングを示す。</p>	<p>特開2003-44093</p> <p>61</p>
<p>(34)</p> <p>62</p> <p>【0266】5. 4. 1 例</p> <p>この例は、電話セッションを操作するために制御イベントに結合 (wire) したスク립ティングを示す。</p>	<p>(33)</p> <p>63</p> <p>【0266】5. 4. 1 例</p> <p>この例は、電話セッションを操作するために制御イベントに結合 (wire) したスク립ティングを示す。</p>	<p>特開2003-44093</p> <p>62</p>
<p>(35)</p> <p>63</p> <p>【0266】5. 4. 1 例</p> <p>この例は、電話セッションを操作するために制御イベントに結合 (wire) したスク립ティングを示す。</p>	<p>(33)</p> <p>64</p> <p>【0266】5. 4. 1 例</p> <p>この例は、電話セッションを操作するために制御イベントに結合 (wire) したスク립ティングを示す。</p>	<p>特開2003-44093</p> <p>63</p>
<p>(36)</p> <p>64</p> <p>【0266】5. 4. 1 例</p> <p>この例は、電話セッションを操作するために制御イベントに結合 (wire) したスク립ティングを示す。</p>	<p>(33)</p> <p>65</p> <p>【0266】5. 4. 1 例</p> <p>この例は、電話セッションを操作するために制御イベントに結合 (wire) したスク립ティングを示す。</p>	<p>特開2003-44093</p> <p>64</p>

67

```

<bind targetElement="textBoxOrigin"
value="//origin_city" />
<bind targetElement="textBoxDest"
test="//sm1/dest_city[&confidence >= 40]"
value="//dest_city" />
</sp:recog>
</t:par>
</body>
</html>

```

【0268】7. SMEX (メッセージ) 要素/オブジェクト 10 が非同期的な、アプリケーション開発者がタイムアウト設定を操作するための内蔵クロックも有する。

【0270】メッセージまたは smex オブジェクトは、通信手段によってアノニミック (anonymous) である。しかし、一実施形態では、smex オブジェクトは、通常の XML やマークアップ要素と同じ寿命を有する。すなわち、smex オブジェクトは、それをホストするドキュメントをアンロードすると消滅する。多くのケースでは、smex オブジェクトはアンロードされると自動クリーンアップを実行し、通信リソースを解放することができるが、マークアップページ間で永続的な通信リンクが望ましい使用事例 (例えば呼の制御など) もありうる。そのような事例のために、このアプリケーションチャチャでは、割り振られたリソースを解放する (例えばソケットを閉じるなど) 責任をアプリケーション開発者に課す。

【0271】smex オブジェクトは、メッセージのフォーマット (スキーマ) についてはニュートラルである。実施形態によっては、既存の標準的なメッセージフォーマット (例えば SOAP または CCXML で使用するスキーマをサポートすることを必要とするのが望ましい) をまぎれ優先して、実装者がいくつかの基本的な場合もある。基本的には、このアプリケーション開発者の両者が、XML あるいはそれに類似のマークアップの規格化された拡張性を最大限に活用して、一方では相互操作性を失うことなく他の機能を導入することを可能にする。

【0272】

例1: ロギングオブジェクトとしての smex の使用

```

<smex_id="logServer">
  <param name="d:server">
    xmlns:d="urn:microsoft.com/COM"
    <d:protocol>COM</d:protocol>
    <d:clsid>209309302020202020942098432098</d:clsid>
    <d:iid>90903859304903498530805309094803</d:iid>
  </param>
  </smex>
</listen...>
...//recog結果を入力フィールドにバインドする他のダイレクティブ
..... <bind targetElement="logServer"

```

(37)

71

ッセージは、次の onReceive イベントが送られるのできる状態になるまで、value として使用することができる。

【0281】・timer: 読み取り/書き込み。タイムアウトイベントをトリガするまでの時間を表すミリ秒単位の数。クロックは、このプロパティに正の値が割り当てられると刻時を開始する。この値は、カウントダウンの進行中に変更することができる。ゼロまたは負の値にすると、タイムアウトイベントをトリガせずにクロックを停止する。デフォルトは 0、すなわちタイムアウトなしである。

【0282】・status: 読み取り専用。オブジェクトの最近のステータスを表す整数。可能な値は、0、-1、および -2 であり、それぞれ、正常、タイムアウトの終了、およびプラットフォームとの通信を確立できない、あるいは通信の中断を意味する。受信されるプロパティを通じて、プラットフォーム固有のエラーメッセージを伝達するとよい。エラーメッセージの伝達が成功した場合、ステータスコードは 0 になる。

【0283】7. 2 イベント  
このオブジェクトは以下のイベントを有する。

【0284】・onReceive: このイベントは、プラットフォームメッセージが到着すると送られる。バインド要素によって宣言されたダイレクティブがある場合には、このイベントを発生させる前にそのダイレクティブを先に評価する。イベントを送る前に、受け取ったプロパティを更新する。

【0285】・onError: このイベントは、タイムアウトが経過したとき、あるいは通信リンクエラーに遭遇したときに送られる。このイベントを送る際、上記のように、ステータスプロパティをそれに対応するエラー \*

```

<smex id="logServer" onload="addFunction()">

```

```

</smex>

```

```

<script>

```

```

function my_logMessage(logClass, message) {

```

```

logServer.send = logClass + "!" + message;

```

```

}

```

```

function addFunction() {

```

```

logServer.prototype.logMessage=

```

```

my_logMessage;

```

```

}

```

```

</script>

```

```

</script>

```

よりオブジェクト指向的な方式でこの情報を参照することができる。

```

logServer.logMessage(RECOG_ERROR, "My message");

```

上記の例のように拡張を機能させるために、smex オブジェクトの実装者にはより多くの作業が要求されるが、すべての必要な機能はすでに確立された規格であることに留意されたい。

【0289】

【発明の効果】以上、説明したように、本発明によれば

【図面の簡単な説明】  
【図1】本発明の実施形態の、コンピュータングデバ

72

\*コードによって更新する。

【0286】7. 3 子要素  
ある要素の形を仮定するとき、smex は以下の子要素を有することができる。

・bind: ダイレクティブを受信メッセージに作用させる点を除いては、reco の場合と同様。

・param: reco の場合と同様。smex オブジェクトのプラットフォーム固有のパラメータを提供する。

各 param 要素は、[name] 属性を使用して名前をつけることができ、param 要素の内容がそのパラメータの値になる。一実施形態では、この要素は、ネームスペースの標準的な XML 属性と XML データタイプ宣言を理解しているべきである。

【0287】7. 4 その他の補足説明  
ロギング機能のために SMEX を拡張する簡潔な方法の1つが以下である。

```

<smex id="logServer"...> ... </smex>

```

```

<script> function logMessage(logClass, message){

```

```

logServer.send = logClass + "!" + message;

```

```

} </script>

```

これは、実際に、その振る舞いを個別に設定することのできる (グローバル) 関数でこのオブジェクトを拡張している。上の例では、ID とメッセージの間にフィールド区切り文字「!」を挿入するようにロギング関数をプログラムしている。

【0288】グローバル関数を好まない者は、ECMA Script の [prototype] プロパティを使用して、この関数をオブジェクトメソッドとして付加することができる。例えば、

```

logServer.prototype.logMessage = function(logClass, message){

```

```

}

```

```

} </script>

```

これは、実際に、その振る舞いを個別に設定することのできる (グローバル) 関数でこのオブジェクトを拡張している。上の例では、ID とメッセージの間にフィールド区切り文字「!」を挿入するようにロギング関数をプログラムしている。

【0289】グローバル関数を好まない者は、ECMA Script の [prototype] プロパティを使用して、この関数をオブジェクトメソッドとして付加することができる。例えば、

```

logServer.prototype.logMessage = function(logClass, message){

```

```

}

```

```

} </script>

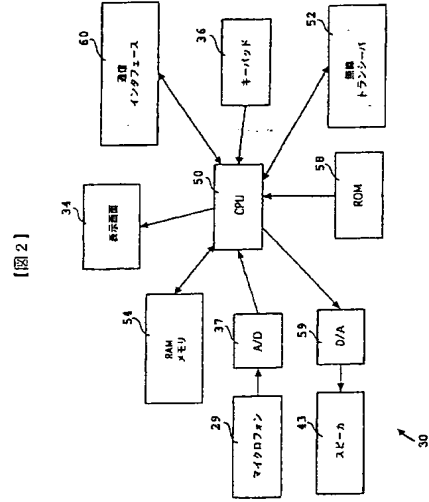
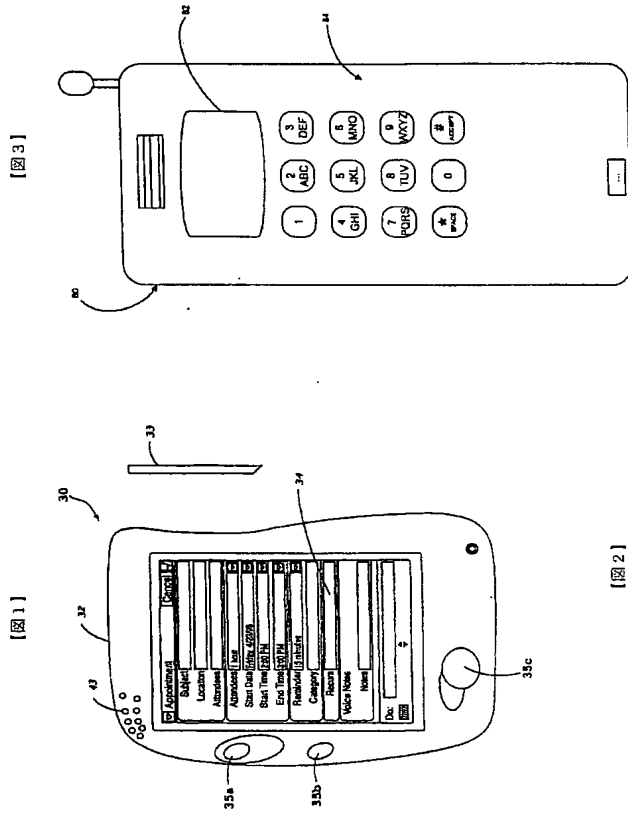
```

75

214	メディアサーバ
216	音声ブラウザ
250、252、254	フィールド
260、270、300	本体部分
262、272、302	スクリプト部分
264	提出ボタン
280、282、284、405	コード部分
281	スケジュール
283、305	読書の開始

76

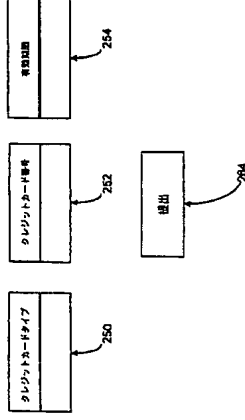
285 音声の検出  
287 音声の終了  
289、291、293、295、297、299、301、303、305  
属性(期間、イベント)  
290、303 コントロール  
307 音声変換システム  
309 パーサ  
320、324 モジュール



-39-

-41-

【图6】



【図 7】

```
<html>
  <form id="get_card_info" method="post" action="http://payment.asp">
    <select name="card_type">
      <option value="MasterCard">MasterCard </option>
      <option value="Visa">Visa </option>
      <option value="ms">MasterCard </option>
    </select>
    <input type="text" name="card_num" width="30"
      onChange="handle()" />
    <input type="text" name="expiry_date" />
    <input type="submit" value="Submit" onClick="verify()" />
  </form>
</html>
<script><[[CDATA[
function handle() {
  if (get_card_info.card_type.value == "amex") {
    if (get_card_info.card_num.length != 15)
      alert ("amex should have 15 digits");
    else
      if (get_card_info.card_num.length != 16)
        alert ("visa and master should have 16 digits");
  }
  function verify() {
    var flag = window.confirm("submit the credit card info?");
    if (flag)
      get_card_info.submit();
  }
}]]>
</script>
</html>
```





前記レコグナイズはジェスチャレコグナイズを含み、前記文法はジェスチャ認識に関連することを特徴とする請求項1に記載のコンピュータ読み取り可能な記録媒体。

【請求項7】

前記レコグナイズは視覚レコグナイズを含み、前記文法は視覚認識に関連することを特徴とする請求項1に記載のコンピュータ読み取り可能な記録媒体。

【請求項8】

クライアント/サーバネットワークにおける音声認識の方法であって、入力音声を表すデータと、認識を行うために、前記入力を表すデータに使用する文法の指示とを、ネットワークを介して受信する受信ステップと、レコグナイズとともに前記文法を使用して前記データを処理し、認識結果を得る処理ステップと、

前記入力を表すデータについての前記認識結果を、前記ネットワーク上の遠隔位置に送信する送信ステップと

を備えたことを特徴とする方法。

【請求項9】

前記指示は、前記文法の位置へのリファレンスを提供することを特徴とする請求項8に記載の方法。

【請求項10】

前記指示は、認識用の言語へのリファレンスを含むことを特徴とする請求項8に記載の方法。

【請求項11】

前記遠隔位置にプロンプトを提供する提供ステップをさらに備えたことを特徴とする請求項8に記載の方法。

【請求項12】

プロンプトを提供する提供ステップは、テキストデータを音声データに変換するステップと、該音声データを前記遠隔位置に提供するステップとを含むことを特徴とする請求項11に記載の方法。

【請求項13】

クライアント/サーバシステム中のクライアントデバイスで実行するためのマークアップ言語を有するコンピュータ読み取り可能な記録媒体であって、

前記マークアップ言語は、非表示式、音声入力ベースのクライアントデバイス、およびマルチモーダルベースのクライアントにおける、認識関連イベント、GUIイベント、および電話イベントのうち少なくとも1つを、前記クライアントデバイスの各々と対話するウェブサーバのために、統一する命令を備えたことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項14】

前記マークアップ言語は、HTML、XML、cHTML、XML、およびWMLの1つを含むことを特徴とする請求項13に記載のコンピュータ読み取り可能な記録媒体。

【請求項15】

前記マークアップ言語は、スクリプティング言語を含むことを特徴とする請求項13に記載のコンピュータ読み取り可能な記録媒体。

【請求項16】

前記マークアップ言語は、同期化マルチメディアマークアップ言語を含むことを特徴とする請求項13に記載のコンピュータ読み取り可能な記録媒体。

【請求項17】

前記マークアップ言語はスクリプティングを模倣することを特徴とする請求項13に記載の方法。